Intro: Widescreen vs Fullscreen

Widescreen vs Fullscreen

Many movies today are filmed with an aspect ratio of 16:9 (width:length) to allow for wider shots. Movies filmed in this format are generally referred to as **widescreen**. However, most older televisions have an aspect ratio of 4:3, and most television programming is presented in the 4:3 format.

This discontinuity in aspect ratio presents a problem to TV programming stations as well as the movie industry when trying to display movies on standard TVs. The two most popular solutions are:

1) Letterbox

Include the entire 16:9 frame inside the 4:3 frame, resulting in black bars at the top and bottom of the screen. This method preserves all of the information from the original frame, but results in a large portion of the 4:3 display being used to display no information.

2) Crop

Crop the sides of the 16:9 frame to generate a 4:3 frame that completely fills the standard TV frame. This method results in a **fullscreen** movie that completely fills the 4:3 display.

However, this method loses a substantial portion of the information from the original movie frame, potentially eliminating important scene elements.

The following modules will focus on developing an improved version of solution number 2 by computationally finding the region of interest in the scene and preserving this information when cropping.

Problems with Cropping

Widescreen to Fullscreen Conversion Loses Information

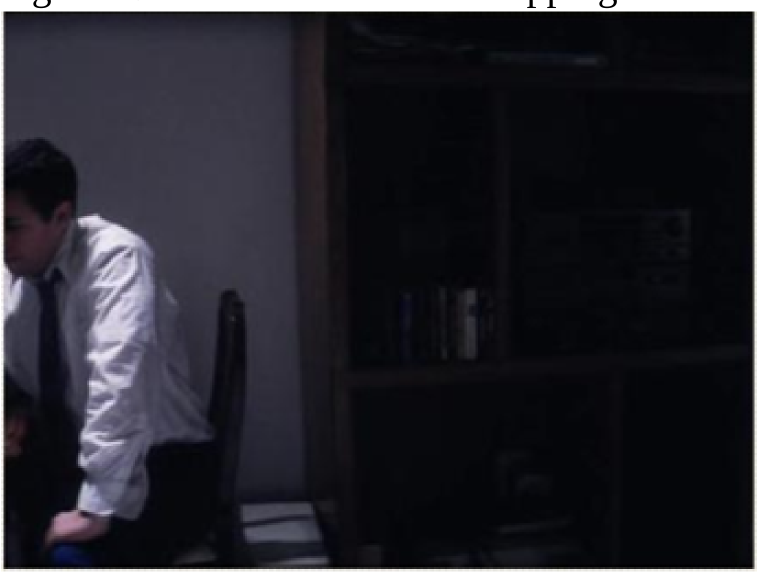
When cropping a widescreen image down to a fullscreen aspect ratio, information will be lost from the edges of the scene. When performing this cropping, there are generally two approaches that are common in the movie and television industries:

1) Center Cut

Using this method, a 4:3 region is defined at the center of the widescreen scene, and the sides are cropped. This results in equal amounts of information being lost from the left and right edges of the scene.

This would not be a huge problem if everything of interest in the movie took place in the direct center of of the scene. However, as you may notice when watching your favorite flick, oftentimes the **Region of Interest (ROI)**is skewed towards the left or right of the scene. When this happens, the center cut method will result in important parts of the scene being cropped out. As seen below in a scene from the film Punch Drunk Love (directed by P.T. Anderson and starring Adam Sandler), a simple center cut approach results in the main character’s being partially cropped out while the majority of the preserved scene contains nothing all that interesting.
Figure 1: Original, widescreen format scene from Punch Drunk Love

Figure 2: Result of center cut cropping



Note how part of the character's face and body has been cropped out

2) Pan-and-Scan

In the pan-and-scan method, an editor goes through the movie and moves around the ROI such that the important elements of each scene are not

cropped out. This is a time-intensive and subjective process, and the results of it will vary depending on who determines the ROI. However, this is the preferred method because it ensures that embarrassing results such as in Figure 2 above do not ocurr.

Better Solution is Needed

What is needed is an automated, quantitative Pan-and-Scan system that can analyze a movie, determine where the important scene elements are, and adjust the ROI so that these important elements are not cropped out.

To develop such a system, we first need to know what the "important scene elements" are so that we can find a suitable method of quantifying them.

What is Important in a Movie Scene?

Motion, Edges, and Focus Determine ROI

Three major indicators of what is important in a movie scene are the amounts of motion, edges, and focus in different regions of the screen.

Motion

Motion is perhaps the best indicator of where the “action” is taking place in a scene. In an uncropped scene, objects that are in motion tend to draw the eyes of the audience. This makes motion detection a good scene element to quantify for our adaptive ROI system.

Edges

Sharp differences in an image tend to indicate the boundaries of separate objects in a scene. Generally, objects that are in focus tend to have more clearly defined edges while objects that are not in focus will have less defined edges. Sharp edges draw the eyes of the audience in a scene, as they are the delineators of separate objects. Thus, by detecting edges in a scene, we can begin to identify which objects stand out from their surroundings and should be included in the ROI.

Focus

What is in focus and what is out of focus in a scene could be said to be the most important way in which a movie director will tell the audience what is important and what they should be looking at. Generally speaking, areas of the scene that are in focus are the areas in which the important action in a scene will be taking place, so detecting the relative focus in different regions of the scene will be an important factor in determining the ROI.

Quantifying the Scene Elements

Now that we have decided on which scene elements we want to include in the ROI, we must develop systems for calculating the “amount” and location of each element in a scene.

Motion Detection via Frame Subtraction

Change in Pixel Values Indicates Motion

A simple method of subtracting one movie frame from another will provide information about which parts of the scene have changed (generally due to motion). This method was performed on each frame of the movie, with consecutive frames being subtracted from each other.

Frame Subtraction

First, the scene is converted to an array of pixel values. These pixel values are the averaged **Red, Green, and Blue (RGB)** values for each pixel. The pixel values of the previous frame are then subtracted from the current frame's pixel values, and the absolute value of the values is taken. The result is an array of values that represent how much each pixel has changed between the two frames, with higher values representing more change. The amount of change in a region of pixels can be interpreted as the amount of motion that is taking place in that region. These data can then be used to determine where in the scene the most motion is taking place.

Illustrative Example

The images below show the results of subtracting two frames from Punch Drunk Love. Note that these are **NOT** consecutive frames, as the changes between consecutive frames can be very small. The frames presented below were chosen to clearly illustrate the results of frame subtraction. In the "difference" frame, higher values (more change) are represented by white.
Figure 1: The current frame


Figure 1: The previous frame (not consecutive frames)

Figure 3: The “difference” frame

White regions indicate more change in pixel values. Very little change has occurred in much of the scene. Only the movement of the main character has resulted in a large amount of change in pixel values. Note that in this scene, the motion is taking place not in the center of the frame, but slightly offset to the left.

Edge Detection via 2D Convolution

Edges in an image are regions of sharp change, often at the boundaries between objects. One way to quantitatively find edges in an image is to analyze the pixel values of an image and examine the gradient of the pixel values matrix. Edges are identified as maxima in the gradient of an image. By using 2-Dimensional convolution and the pixel value matrix for each frame from a movie, the gradient can be calculated and used to find the edges.

2D Convolution

For our adaptive ROI system, the pixel value matrix was convolved with two different matrices (h_1 and h_2) using the following formula:

Equation:

$$\nabla x = \sum_{i=0}^{k1} \sum_{j=0}^{k2} x(i,j)h(m-i,n-j)$$

Where:

Equation:

$$\nabla x = \text{gradient}$$

Equation:

$$x = \text{image}$$

$$h_1 = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} \text{ and } h_2 = \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$$

The result of convolution with h_1 gives the horizontal gradient of the frame while convolution with h_2 gives the vertical gradient. These are then combined to find the magnitude of the gradient at all points in the frame.

Equation:

$$| \nabla x | = \sqrt{(\nabla x_{\mathrm{HORIZ}})^2 + (\nabla x_{\mathrm{VERT}})^2}$$

The gradient magnitudes are then thresholded, and any gradient magnitude greater than the threshold is recognized to be an edge. Thus the result is a matrix of zeros and ones, where a 1 indicates that that pixel is part of an edge, and a 0 indicates it is not part of an edge. For our system, a threshold of 125 gives satisfactory results and recognizes only sharp edges.

Illustrative example

Below you can see a frame from Punch Drunk Love as well as the result from the edge detection method. For illustration purposes, the detected edges are displayed in white. The four vertical white lines are there to show the different regions of the screen that are eventually used to decide which region has the most edges in it.

Original frame

Detected edges

White indicates an edge. The vertical bars are illustrative of how the frame could be segmented for analysis. Note that the area of interest is far to the left of the frame, not in the middle.

Focus Detection

Focus detection

Focus Detection:

One important aspect of images is focus. While qualitatively deciding whether an image is in focus or not is relatively easy, quantitatively it can be quite difficult. One way to detect whether or not an image is in focus is by examining its power spectrum.

Power Spectrum and Focus

It is generally assumed that natural images are made up of fractals, and it can be shown that the **power spectrum** (power as a function of frequency) of a natural image should fall off as
Equation:

$$\frac{1}{f^2}$$

where f is the frequency.

As an image goes out of focus, it becomes blurred. That is to say that the edges are less sharp. If an image contains less sharp edges, its power spectrum will contain less high-frequency power. The power spectrum of an out-of-focus image should, therefore, fall off faster than an in-focus image.

So by calculating the power spectrum and examining its linear regression on a loglog plot (log[power] vs log[frequency]), we can get an indicator of focus.

Calculating the Power Spectrum

The power specturm is simply the square of the two dimensional Fourier transform:

Equation:

$$P(k_x,k_y) =\mid F(k_x,k_y)^2 \mid$$

where the two dimensional Fourier transform is given by:
Equation:

$$F(k_x,k_y) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(k_x,k_y)^2 e^{\frac{-j2\pi}{N}\left(\mathrm{xk}_x+\mathrm{yk}_y\right)}$$

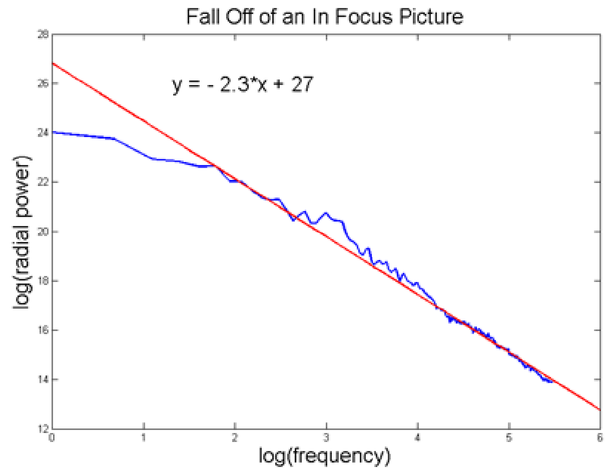
Note that denotes an individual image pixel. You may have noticed that the above equations define a square image. While a non-symmetric two dimensional Fourier transform exists, using square images eases the process.

Because whether or not an image is in focus depends on the magnitude of power as a function of frequency, once the two dimensional power spectrum is computed as above, we **radially average** the spectrum. That is, the average of the values which lie on a circle a distance R from the origin is taken. Because frequency increases linearly in all directions from the origin, radially averaging the power spectrum gives the **average power at one frequency**, effectively collapsing the two dimensional spectrum to one dimension. It should be noted that $F(k_x,k_y)$ has been centered around baseband, meaning the frequency of the rotionally averaged power spectrum extends from 0 to N/2 -1.

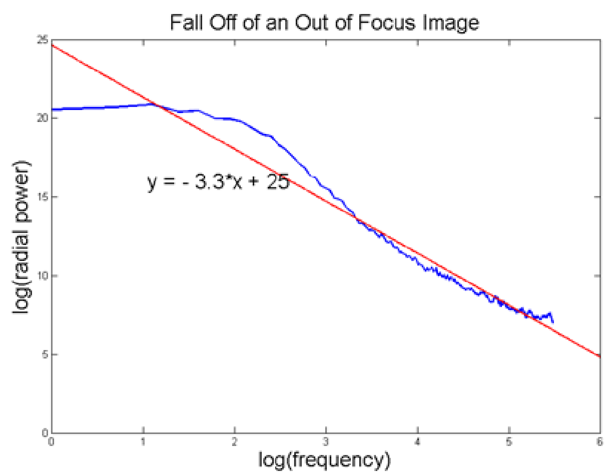
The power spectrum’s falloff on a loglog plot can now be examined to determine focus.

Illustrative Example of Focus Analysis on Entire Image

The following images show the results of a linear regression of the power spectrum on a loglog plot for an in-focus image and an out-of-focus image.
Focus analysis of an in-focus image



Focus analysis of an out-of-focus image



As expected, the out-of-focus image yielded a linear regression with a slope of -3.3, while the in-focus image yielded a linear regression with a slope of -2.3, indicating that the out-of-focus image has fewer high frequency components.

Determining Regions of Focus

Because frequency and power should be related exponentially as stated before, the loglog plot should display a linear relationship. Taking the linear regression of the loglog plot leads to an estimate of the frequency fall off.

For example, if the linear regression where to return a slope of -2, we know that the power spectrum falls off as $\frac{1}{f^2}$.

The same principles used to determine whether or not an **image** is in focus can be used to determine what **region** of an image is in focus. Because cameras can only focus on one spatial plane, in a single picture certain objects will be more in focus than others. To determine which region of an image is in focus, one simply has to divide the image into separate spatial region and then use the methods described above on each region. The region whose power spectrum conforms most closely to the $\frac{1}{f^2}$ fall off can be considered the center of focus in the image.

Determining the ROI

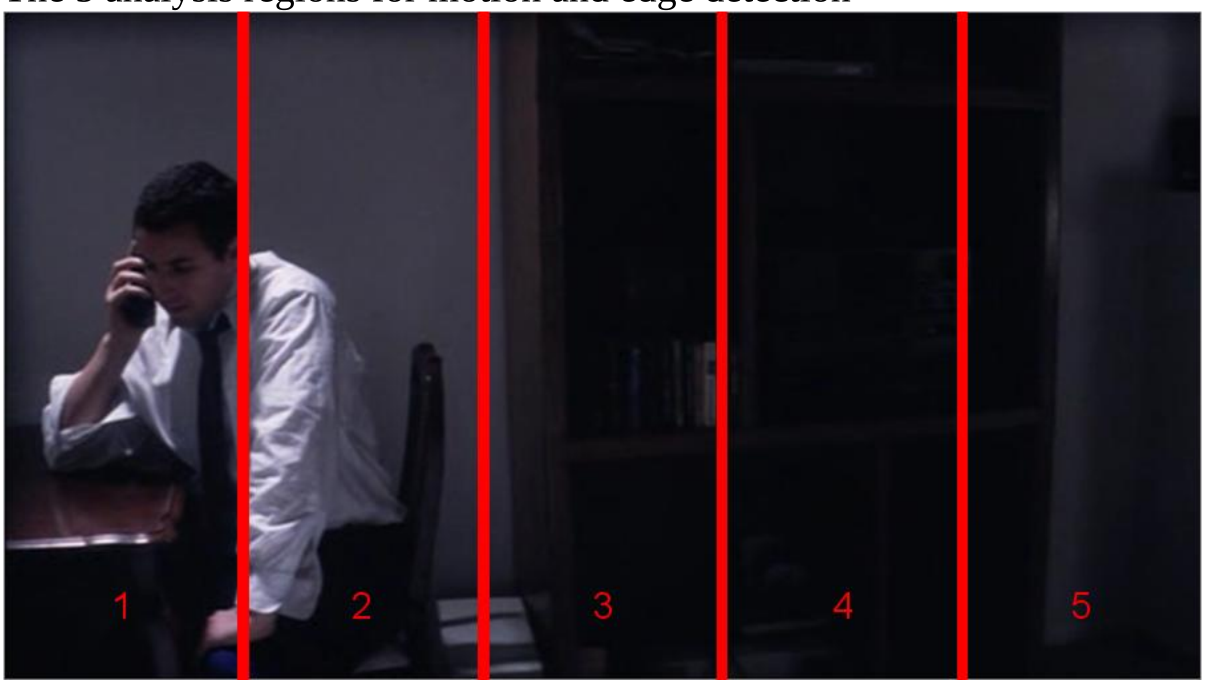
Incorporating Motion, Edge, and Focus Detection

In order to determine the ROI for each frame of a movie, we need to be able to incorporate the results of motion, edge, and focus detection into a single system. The way we accomplish this is by analyzing each frame in five separate sections.

Frame Region Analysis

For motion and edge detection, the entire frame is processed at once, and then the resulting matrix is broken into 5 regions as below:

The 5 analysis regions for motion and edge detection



Scene from "Punch Drunk Love"

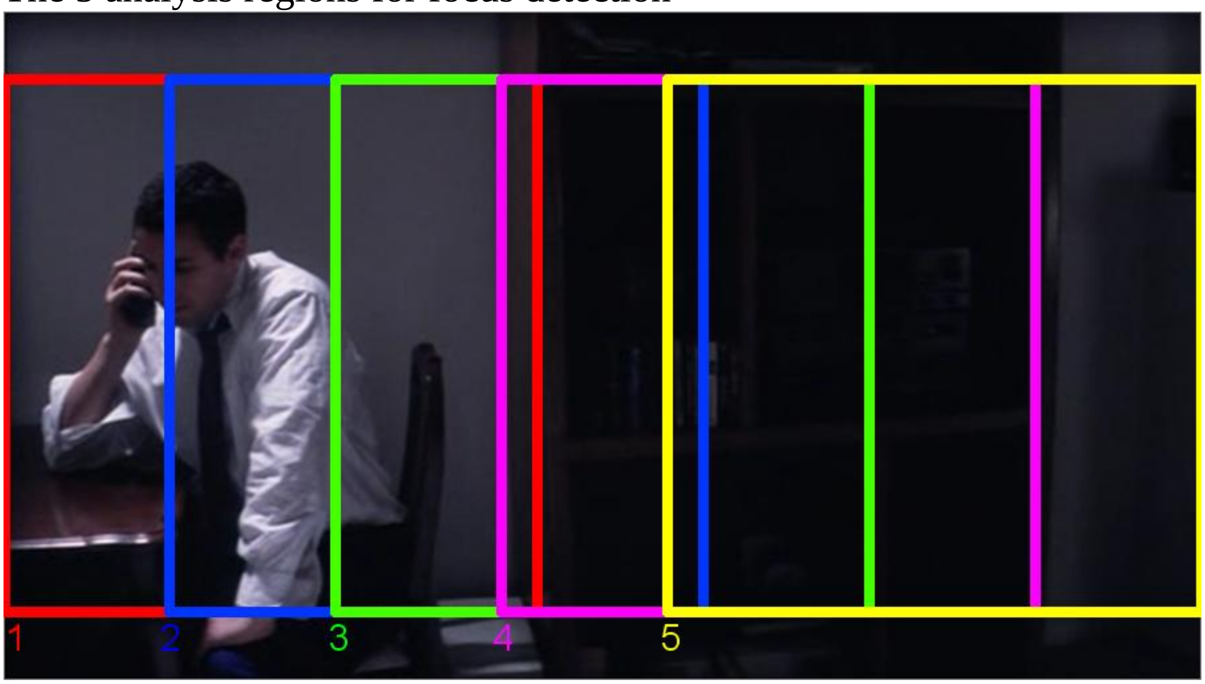
For motion detection, recall that the processing results in a "difference" matrix after subtracting two frames. The mean of the difference values is

taken for each region and divided by the mean of the difference values for the whole frame. These region means are then normalized by the one with the largest magnitude. The result is a number between 0 and 1 for each region, with a value of 1 indicating the region of maximum relative change and 0 indicating no change.

For edge detection, recall that the processing results in an edge matrix, where a value of 1 means that that pixel is part of an edge and a value of zero indicates that the pixel is not part of an edge. Thus the sum of the pixels in each region is found and normalized to the region with the highest sum. The result is a value between 0 and 1 for each region, with 1 indicating the region with the most edges and 0 indicating a region with no edges.

For focus detection, recall that the processing results in a value for the slope of the linear regression of the loglog plot of the power spectrum. Due to the requirement of a square matrix for the 2D Fourier transform, the frame is divided into 5 semi-overlapped square regions:

The 5 analysis regions for focus detection



Scene from “Punch Drunk Love”

The focus detection processing is performed on each of these regions, and then the most in-focus region is identified (by its falloff rate), and the remaining regions are assigned a normalized value corresponding to how close they come to having the best focus value. The result is a value between 0 and 1 for each region, with 1 indicating the region of best focus and 0 indicating the region of worst focus.

Assigning the ROI

After converting the results of motion, edge, and focus detection into values between 0 and 1 for each region, the values are averaged for each region. This gives one value for each region, with higher values indicating that there are more elements of interest in that region.

To translate this into an ROI, a horizontal midpoint is defined within the widescreen frame, and each region's interest value is mapped to a weighted deviation from this midpoint. The net deviation from midpoint is then found by summing these deviations, and the fullscreen ROI midpoint is defined to be at this deviation from widescreen center.

Thus, interest in regions 1 and 2 act to pull the fullscreen ROI to the left, while interest in regions 4 and 5 acts to pull the fullscreen ROI to the right, and activity in region 3 acts to maintain the fullscreen ROI at the center.

The final midpoint value is filtered with a moving average (half-width = 30, or approximately 1 second of video) to eliminate jerky ROI movements.

Adaptive ROI Results

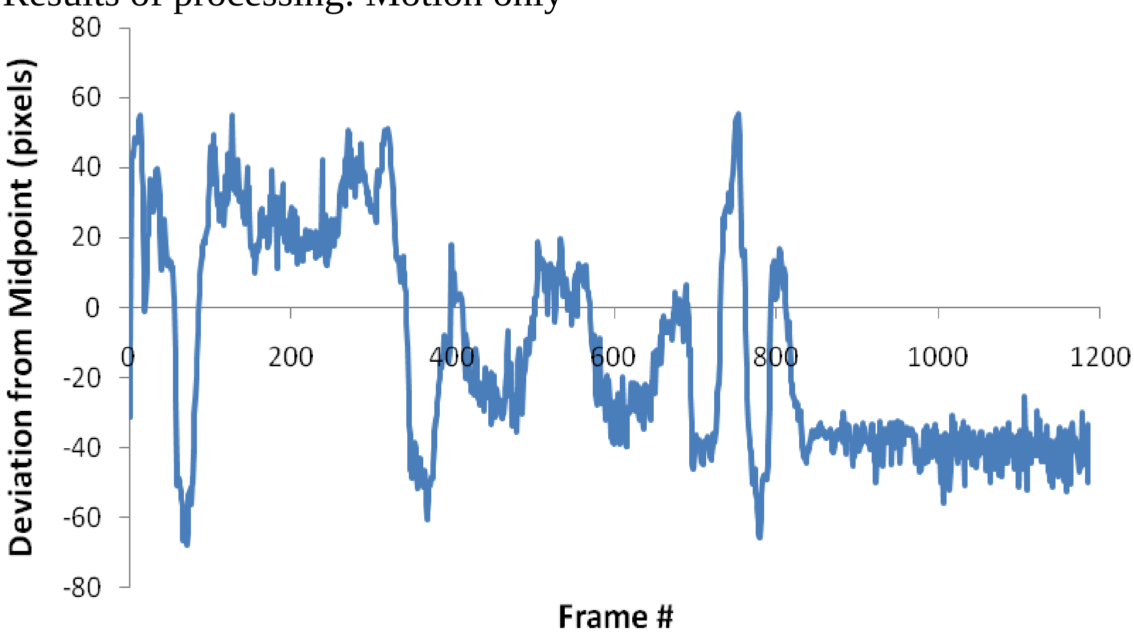
Implementation With a Widescreen Movie

To evaluate our adaptive ROI system, a scene from the movie “Punch Drunk Love” was analyzed. A straight-cut fullscreen version was generated, and our adaptive ROI system was used to generate a pan-and-scan fullscreen version. The pan-and-scan version was analyzed by tracking the ROI midpoint as a function of frame number.

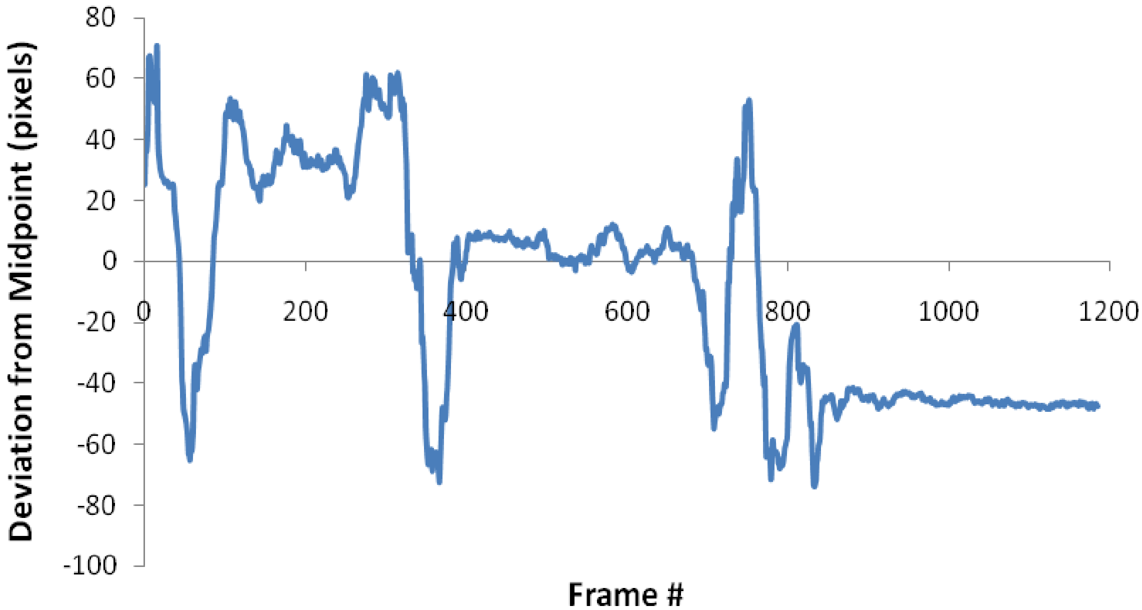
ROI Midpoint Tracking

To illustrate the mobility of the ROI midpoint, the following charts show the deviation of the midpoint from center after processing by motion, edge, and focus detection individually and combined. The first 4 charts show the unfiltered results of processing, while the last chart shows the filtered ROI midpoint using the combined processing.

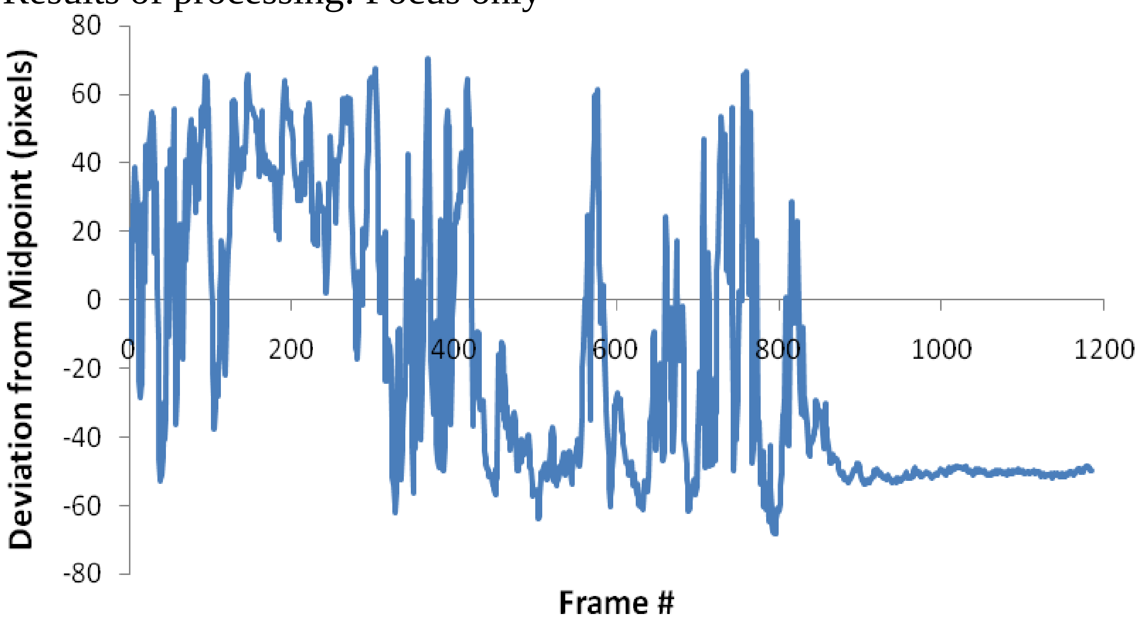
Results of processing: Motion only



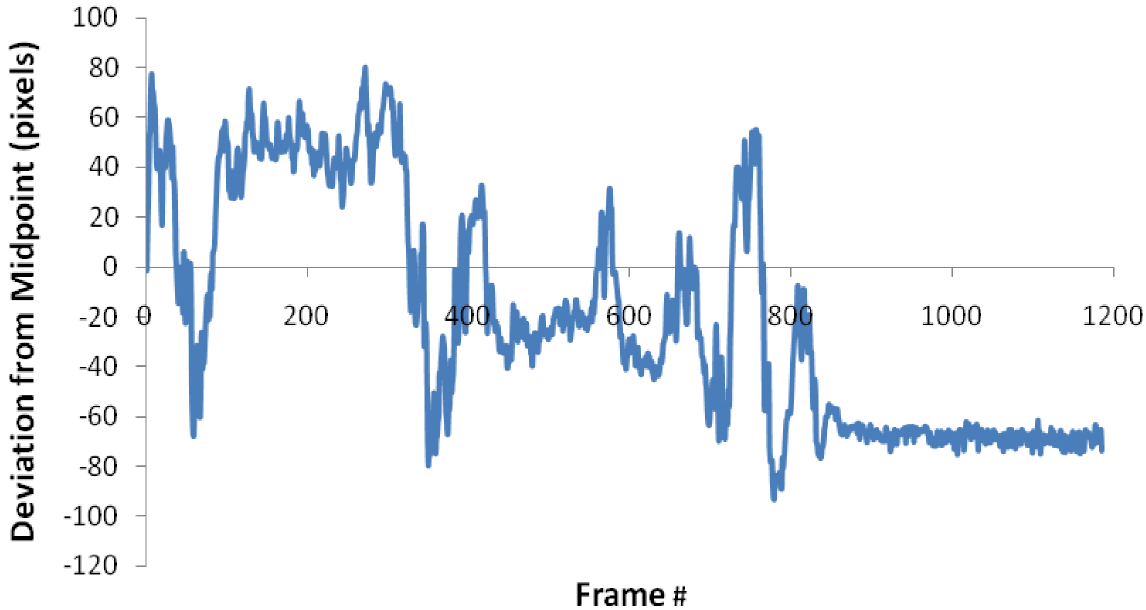
Results of processing: Edges only



Results of processing: Focus only



Results of processing: Combined methods, NO filtering



Results of processing: Combined methods, WITH filtering

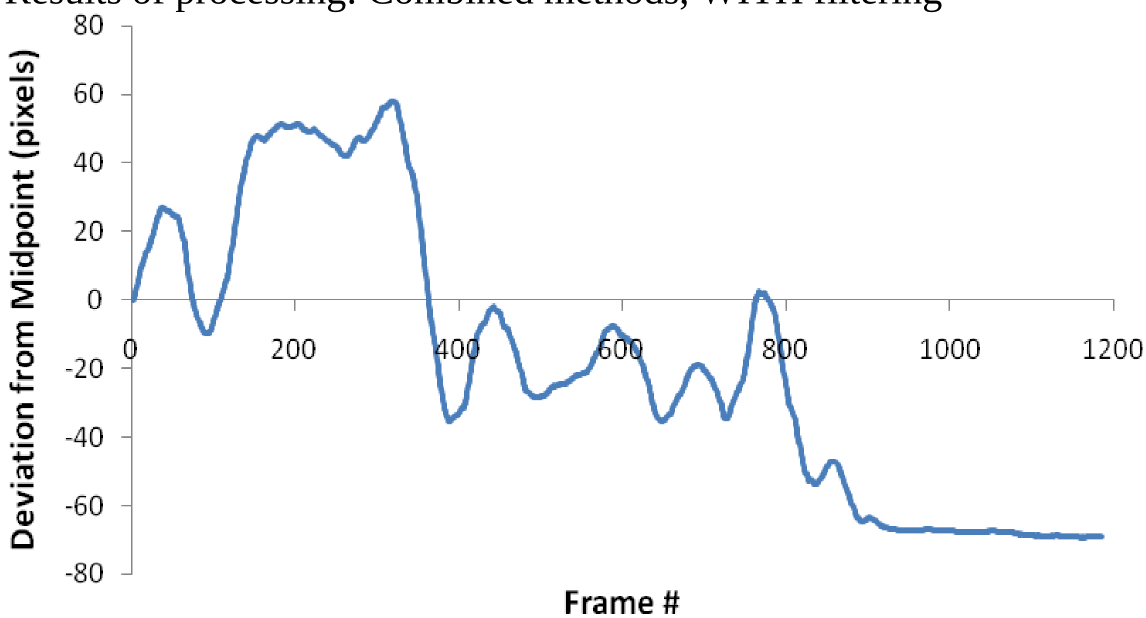
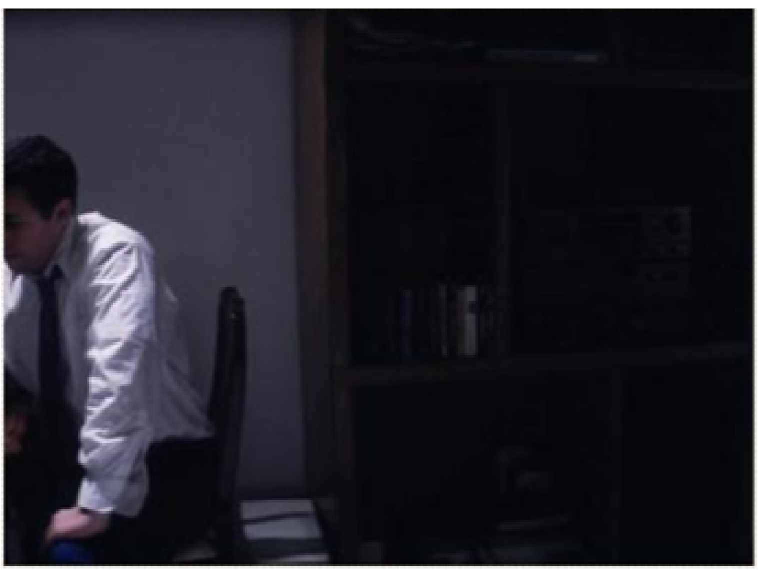
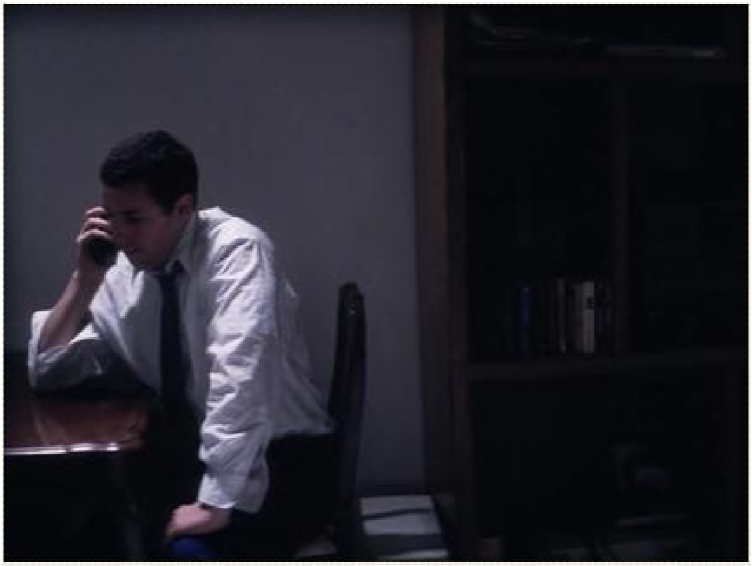


Image Results

The following images illustrate the functionality of our adaptive ROI system compared to a center-cut fullscreen image. The images are taken from the same scene as was used for the above analysis, in the Frame # 1000 to 2000 range.

Original widescreen frame


Result of center-cut fullscreen conversion

Result of adaptive ROI fullscreen conversion



Video Results

Here you can download a center-cut fullscreen and an Adaptive ROI fullscreen version of the scene above. The videos “with black bars” show a 16:9 video with black bars over the regions that would be cropped out – the ROI is still 4:3.

Note that the videos go further in the scene than the charts above.

[missing_resource: Adaptive ROI fullscreen.mp4]

[missing_resource: Adaptive ROI with black bars.mp4]

[missing_resource: Center cut fullscreen.mp4]

[missing_resource: Center cut with black bars.mp4]

Applications, Limitations, and Future Work

Applications

An adaptive ROI system such as this could be used for several things. First and foremost, it could be used in the television industry to quickly and inexpensively modify widescreen movies for fullscreen broadcasting. This would eliminate the problems associated with center-cut cropping and it wouldn’t require someone to spend the time to manually pan-and-scan the fullscreen conversion.

If it could be made to run fast enough, it could be implemented in DVD players and computer-based video players (such as VLC and Windows Media Player) so that the user could determine whether they would like to watch the widescreen or the fullscreen version of their movie, depending on their monitor.

This system could also have applications in any kind of surveillance or monitoring system, as it could save on data storage by recording the regions of interest.

Limitations

There are a number of problems with this system that prevent it from immediate implementation. First and foremost, a better filtering system is needed to smooth out the movements of the ROI, because at times it is quite shaky despite the smoothing filter already being used.

Secondly, the system should be improved to recognize scene and shot changes and reset the ROI to center at these changes, so that the filtering from the ROI of the previous shot does not affect the ROI of the beginning of the current shot.

And finally, there are a lot of ways that the incorporation of the motion, edge, and focus detection results could be optimized. Right now they are simply being averaged, but it is highly likely that there is an optimum

weighting of the three elements that would make for smoother, more eye-pleasing ROI determinations.

Future work

In addition to correcting the limitations of the system, there are other things that could be done. For instance, motion detection could be implemented on the detected edges, thus identifying moving edges. General optimization and improvement of all three of the detection methods could be performed, and better ways of separating foreground interest from backround interest could be implemented. And finally, adaptive seam carving and image restructuring could be performed to try to re-size the widescreen down to a fullscreen movie without cropping out info from just the sides.

LabVIEW VI

The adaptive ROI system developed above was implemented using LabVIEW 2009,

from National Instruments. You can download the file here, but in order to run

the system you will need the Vision and MathScript modules for LabVIEW, as well

as a widescreen .avi to process.

You will also need plenty of storage space, as the .avi that the system saves is

quite large -- you can modify the VI to compress the video if you are familiar

with LabVIEW.

Alternatively, the motion, edge, and focus detection algorithms were implemented

in LabVIEW via mathscript. They are included in an m-file, so you can take these

algorithms and use them as you will.

Files

download mathscript [missing_resource: Adaptive ROI system.vi]